

LKDOS MACHINE LANGUAGE SUBROUTINES

The following information is for directly using the routines within the dos to control the diskdrive.

They are useful in examining or changing data on the disk and also you can create custom files and simulate random access files etc.

To access dos subroutines within the cartridge directly you must use machine language.

To call these subroutines you must first turn ON the cartridge with a Call to address 98(dec). The cartridge is turned OFF by a memory read at address 100(dec).

You should also disable the interrupts before entering and enable them after you are done.

EG: to access the cartridge.

```
DI
CALL 98
CALL dos subroutine
other user code
LD A,(100)
EI
RET
```

DESCRIPTION OF USEFUL SUBROUTINES

The addresses of these subroutines are on the following pages. The jump table of calls is used so that the addresses of the main routines doesn't change when a change or revision is made to the LKdos Eprom. Also because of the size of the dos, it was assembled in 2 modules. The second module uses the table to link to the first module.

SAVEBF - Saves buffer to the current track and side as setup by TRACK or NEXTR. Disk should be spinning for 1 second before using this routine. You can load data to and from the buffer using the LDIR command in machine language.

LOADBF - Loads the buffer from the current track and side. No spinup necessary.

TRACK - Restores the drive head to track 0, then seeks the track held in curtrk. The number in curtrk is actually the block number. Even blocks are side 0, Odd are side 1. The track# is = INT(block/2) EG: Block 11 = track 5 side 1.

NEXTR - Increments curtrk by one and moves the head to the next track or side. The drive should be running before this is called.

INDIR - Sets the head on track0 and loads the directory trk. Then searches for a file name matching the one the user has put at progrm. The temp2 pointer holds the address of the file name if found. Errnu will contain 10 if the file wasn't found. Clear errnu before calling this. The file name should be put at progrm before calling this with spaces following the name to a full 10 characters if the name is short.

WPROT - Does 3/4 second spinup then checks for write protect sticker. If found it prints an error. This routine will stop your program on an error. Better to check it by manually.

DECDP - Prints decimal # held in temp1 and accumulator.

DOSER - This will print a standard type Error the same way that the 2068 does. The HL register should hold the start of the error message and the message should end with a period '.'.

SAVING AND LOADING PROGRAMS FROM MACHINE CODE

You can directly access the main operating system by the following 4 subroutines. This will allow you to write or modify fully machine code programs to use the disk just as you can in basic.

The reason you can't access the dos by the same way basic does is that in basic, the program line contains all the information and the dos scans the line for formulas and file names.

The dos location 'NmiF' is used to tell the dos how it is being called (NMI, Basic or from a machine code program). It should be loaded with 11(dec) before you call the following routines.

SAVING FROM MC CONTINUED

To Save an area or memory to disk from MC, you must give it a file name and transfer the file name to progmn. The file name should be a Code type but since there is no syntax checking, any name and extension will work. You are not limited to .A B or .C type files.

Then call the SV#1 subroutine. This will enter the file name in the directory and fill the dir workarea with free blocks to use for saving.

Now load temp2 with the length of the file and temp4 with the start address of the file. Then call SV#2 to finish the save.

EG: to save a file named 'test.ZZ' start add 30000 length 10000

3.

Save	DI	disable int.
	CALL 98	turn on cartridge.
	LD HL,Name	transfer file name
	LD DE,progmn	to progmn
	LD BC,10	
	LDIR	
	LD A,11	Set MC save.
	LD (NmiF),A	
	CALL SV#1	Save name .
	LD HL,30000	
	LD (temp4),HL	Set start address.
	LD HL,10000	
	LD (temp2),HL	Set length of save.
	CALL SV#2	Second data save data
	LD A,(100)	Exit cartridge
	EI	
	RET	
Name	DEFM /test.ZZ	/ Name of file.Spaced out to 10 characters

The SV#1 mainroutine first calls the WPROT subroutine to see if the disk is protected. If it is, then it will terminate with a error. This will stop a MC program.

If you dont want your MC program to be stopped the best way around this is to find the address of the actual SV#1 routine in the eprom by reading locations 205 and 206 then adding 3 to this to get past the WPROT call, then calling this new address directly. This address must be calculated dynamically because it may change with new revisions of eproms.

LOADING A FILE FROM DISK

Loading a file from MC is similar to saving. Just use LD#1 instead of SV#1 and LD#2 instead of SV#2. The start and length still go into temp4 and temp2. The file name is also moved into place the same way.

You can poke 23728,100 the same way as a basic load, so that you dont halt your program with a File Not Found error.

If you have need any futher info on how to access the dos, let me know.

-LKdos was written on a 2068 using the ZEUS assembler.

7game-
@49153

```

00010 *****
00020 LOCATIONS USED BY THE DOS CALLS
00040 *****
00050 NmiF EQU 8194 Nmi, User or Basic Flag
00060 Dvsel EQU 8195 Drive select byte (1770)
00070 Hspd EQU 8202 Head speed for FD1770
00080 Rbank EQU 8203 Ram Disk bank#
00090 Raddr EQU 8204 Ram Disk bank address
00100 wrkspc EQU 8220 Start of Disk workspace
00110 buffer EQU wrkspc +84 8304 Start of track buffer
00120 data EQU buffer +24 8328 Start of Actual data block
00130 curtrk EQU wrkspc +1 8305 Block # for head to seek
00134 errnu EQU wrkspc +4 8308 0 if no error occurred 270
00135 progm EQU wrkspc +6 8310 Holds string from GTfil
00140 temp1 EQU wrkspc +20 8324 Used by DECDMP
00150 temp2 EQU wrkspc +21 8325 Usually length of file
00160 temp4 EQU wrkspc +23 8327 Start address of file
00170 temp6 EQU wrkspc +25 8329 temporary address
00175 dirwka EQU wrkspc +30 8334 Holds Cell being used
**** These are registers are saved with each track ****
00180 destin EQU buffer +12 8340 address of data block - ?
00190 lenth EQU buffer +14 8342 lenth of data block - ?
00200 Linenu EQU buffer +17 8345 Auto start Basic Line
00210 Vofset EQU buffer +20 8348 Vars Offset (Total-Vars)
00220 Totlen EQU buffer +22 8350 Total lenth of file - ?
*****
Bufferlength Data length
5114 5090 Double Density Controller
1984 1960 Single Density Controller
*****
This is the LKdos main subroutine jump table . Each Call
is 3 bytes apart. These addresses are unaffected by any
Changes or revisions made later to the Dos.
00250 *****
00260 STARTS AT ADDRESS 120 (decimal)
00270 -- CALL-----ADDRESS
00280 SAVEBF JP crcsv 120 Save the buffer to the disk
00290 LOADBF JP loadbf 123 Load the Buffer from the disk
00300 TRACK JP trac 126 Restore to Trk0 then seek curtrk
00310 NEXTR JP nextrk 129 Increment disk head 1 trk or sid
00320 INDIR JP indir 132 Check dir for file (in progm)
00330 MOVDR JP movdr 135 Move cell to dirwka
00340 CMDCK JP cmdck 138 Check command syntax
00350 ENDLN JP Endoln 141 Move CHADD to end of basic line
00360 EVALU JP Evalu 144 Evaluate Numeric formula
00370 NOFIL JP nofil 147 No file error
00380 WPROT JP WPROt 150 Check for protect sticker
00390 ZERO JP zero 153 Restore blocks used by cell
00400 GTFIL JP GTfil 156 Evaluate String , Put in progm
00410 ROMM JP ROMS 159 Check for spectrum rom
00420 NEWET JP newet 162 Put new entry in dir
00430 DECDP JP decdmp 165 Print temp1 in decimal
00440 TRANOK JP tranok 168 Final routine for save
00450 DOSOP JP DOSOP 171 Close the disk channel
00460 DOSer JP Doserr 174 Print Error HL holds Messag
00470 CLRBF JP clrbf 177 Clear buffer
00480 ENCBF JP encdbf 180 Encode the buffer with Addresses
00490 VSERCH JP vsersch 183 Look for Arrays
00500 JPOUT JP GTOUT 186 Exit cartridge
00510 GROW JP Grow 189 Insert space in program
00520 SHrnk JP Shrink 192 delete space in program
00530 DTERR JP Fatal 195 Cat data error
00540 LD#1 JP Lsubr 198 First half of User load command
00550 LD#2 JP LDdata 201 second half
00560 SV#1 JP Ssubr 204 First half of User save command
00570 SV#2 JP SMEM 207 second half

```

NOTE- Most of the cartridge ram is used by the dos and ex-bas
operating system, But the merge buffer can be used for temp
storage . It follows the data buffer and is 1200 bytes long.